Scaler une application shiny

Cervan Girard 1*

Mots-clefs: Shiny – Scalabilité – API – Base de données

Résumé

Ces dernières années, `{shiny}` a trouvé sa place dans le vaste monde du développement d'applications web. Cependant, là où les outils de création et d'extensions font légion, le déploiement à grande échelle reste une thématique intimiste, apparaissant parfois comme une compétence réservée à une poignée d'initié. Pourtant, les outils existent, et n'attendent que nous pour être utilisés. C'est ce que nous verrons dans cette présentation : comment créer et déployer une application `{shiny}` à grande échelle, et ce sans l'utilisation de RStudio Connect ou de Shinyproxy. Comment ? D'abord, en pensant "scaling" dès le développement, notamment en permettant l'utilisation de plusieurs sessions R (et donc plusieurs personnes) sur une même instance de l'application. Pour cela, nous prendrons quelques exemples comme l'utilisation d'un base de données externe, l'intégration d'une API REST, l'intégration de `{future}` et `{promises}`, ainsi que le caching. Pourquoi ces modifications ? Pour déporter au maximum le travail de R, et donc de `{shiny}`. À termes, cette application devient un "passe-plat", qui a pour objectif d'interroger une API ou une base de données, laissant uniquement à R les actions pour lesquelles il est bon : graphiques, modélisations, etc. Nous verrons que ces processus doivent être eux aussi optimisés, afin de pouvoir gérer la charge de plusieurs personnes en simultanée. Ensuite, une fois que l'application est optimisée, vient le moment de la déployer à grande échelle. Nous verrons comment déployer cette application sur plusieurs serveurs différents, et comment nous pouvons dispatcher la charge en distribuant les utilisateurs sur les différentes instances de l'application.

Des exemples sont disponibles sur le repos suivant :

- https://github.com/Cervangirard/scalingshiny
- https://github.com/Cervangirard/bddshow

*ThinkR, cervan@thinkr.fr